AMENDMENTS TO THE CLAIMS:

1. (CURRENTLY AMENDED) A method for facilitating <u>autonomous</u> modification of a hardware state of a fixture system, comprising:

<u>a firmware of the fixture system monitoring hardware state changes of a fixturing device of the fixture system;</u>

<u>in response to the firmware receiving a hardware state change of the fixturing device corresponding to a programmable event stored in a memory of the fixture system, the firmware autonomously</u> triggering <u>execution of</u> a macro of one or more compiled macros <u>corresponding to the programmable event</u> ~~in response to one or more stimuli~~, ~~wherein:~~

~~said one or more compiled macros are created using a high-level programming macro language;~~

~~compiling one or more macros into a format recognizable by an interpreter residing within a fixturing device;~~

~~transferring the one or more compiled macros to a firmware residing within the fixturing device;~~ and

the firmware running the triggered macro and executing one or more commands contained therein ~~in response thereto, thereby facilitating modification of the hardware state of the fixture system comprising the fixturing device and the one or more macros that allow the fixture system to interact with a product under test, wherein the fixturing~~

~~device is operable to hold the product under test stationary and to support motions that allow for connection to the product~~.

2. (PREVIOUSLY PRESENTED) The method of claim 1, wherein each of the one or more commands are interpreted sequentially.

3. (PREVIOUSLY PRESENTED) The method of claim 1, wherein the high level macro programming language may be determined by the fixturing system.

4. (ORIGINAL) The method of claim 1, wherein the one or more macros are compiled external to the fixturing device.

5. (ORIGINAL) The method of claim 1, wherein prior to the firmware interpreting the triggered macro, a triggered macro byte code is transferred to a local memory of the fixturing device.

6. (ORIGINAL) The method of claim 1, wherein the macro is triggered by one or more internal events corresponding to one or more hardware states of the fixturing device.

7. (ORIGINAL) The method of claim 6, wherein the one or more internal events are stored in a nonvolatile memory of the fixturing device.

8. (ORIGINAL) The method of claim 1, wherein the macro is triggered by one or more external commands transmitted by a control software module.

9. (ORIGINAL) The method of claim 8, wherein the control software module is a compiler for the one or more macros.

10. (ORIGINAL) The method of claim 1, wherein the one or more macros are compiled into byte code.

11. (ORIGINAL) The method of claim 10, wherein the byte code is downloaded into a nonvolatile memory of the fixturing device.

12. (ORIGINAL) The method of claim 11, wherein one of a revision code is downloaded with the byte code, said revision code operable to determine a version of one or more macros currently loaded within the fixturing device.

13. (ORIGINAL) The method of claim 12, wherein during a system initialization, further comprising:

a control software comparing a first macro revision with a second macro revision determined by a default macro file; and

if the first macro revision and the second macro revision are not equivalent, the control software compiling and downloading the one or more macros from a file.

14. (CURRENTLY AMENDED) A structure that facilitates a modification of a hardware state of a fixturing device <u>autonomously</u>, further comprising:

a supervising automation software module, coupled to a control software module of a computer program product, said automation software module operable to initiate operation of the structure; and

a fixturing device, coupled to the control software module, ~~operable to hold a product under test stationary and to support motions to allow for connection to the product under test,~~ said fixturing device further comprising:

a firmware module, said firmware module operable to receive one or more stimuli, ~~preferably~~ corresponding to one or more hardware state changes of said fixturing device; <u>and</u>

one or more local memory modules, coupled to the firmware module, said local memory modules operable to ~~contain~~ <u>store</u> one or more compiled macros and <u>one or more programmable events</u> ~~the one or more stimuli preferably corresponding to the one or more hardware states; and~~

~~the one or more compiled macros,~~ <u>wherein</u> in response to the firmware module receiving the one or more stimuli <u>that correspond to a programmable event of the one or more programmable events</u>, <u>the firmware module autonomously triggering execution of a macro of the one or more compiled macros corresponding to the programmable event that causes</u>

~~causing~~ the one or more hardware states of the fixturing device to be changed.

15. (ORIGINAL) The structure of claim 14, wherein the one or more stimuli are events receivable by the firmware module.

16. (ORIGINAL) The structure of claim 14, wherein the one or more stimuli are commands receivable by the firmware module.

17. (ORIGINAL) The structure of claim 14, wherein the control software module is coupled to the fixturing device via an electronic transmission cable.

18. (ORIGINAL) The structure of claim 14, wherein one or more of the one or more local memory modules are nonvolatile.

19. (PREVIOUSLY PRESENTED) The structure of claim 14, wherein the firmware module is operable to change the one or more hardware states in response to the one or more stimuli.

20. (ORIGINAL) The structure of claim 14, wherein the one or more compiled macros were previously compiled using the control software module.

21. (PREVIOUSLY PRESENTED) The structure of claim 14, wherein the one or more compiled macros are operable to be interpreted during an operational mode of the fixturing device.

22. (ORIGINAL) The structure of claim 14, wherein the control software module sends one or more commands, receivable by the firmware.

23. (PREVIOUSLY PRESENTED) The structure of claim 22, wherein the firmware module, upon receiving the one or more commands, executes one or more of the one or more compiled macros contained within the one or more local memory modules.

24.   (NEW)   The method of claim 1, wherein said one or more compiled macros are created using a high-level programming macro language.

25.   (NEW)   The method of claim 1, further comprising:

compiling one or more macros into a format recognizable by an interpreter residing within a fixturing device; and

transferring the one or more compiled macros to the firmware residing within the fixturing device.